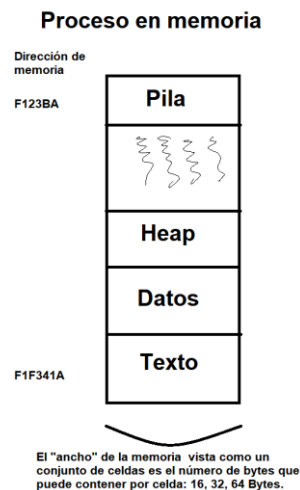


Administración de procesos

Un **proceso** es un programa en ejecución. Este incluye la actividad actual por medio de un contador de programa y el contenido de los registros del procesador. Cada proceso cuenta con una pila, en la cual, se alojan datos temporales: parámetros de una función, variables locales, etc.; y con una sección de datos que contiene las variables globales.

Además, a cada proceso se le asigna una porción de memoria llamada heap, la cual, es asignada de manera dinámica (durante el tiempo de ejecución).

Otra sección de memoria asignada al proceso es el llamado texto. Este contiene las instrucciones del programa.



Diferencia entre proceso y programa

La diferencia entre programa y proceso puede comprenderse mejor con una analogía. Supongamos que un chef va a cocinar un pastel. Para ello, cuenta con una receta, una cocina, y todos los ingredientes-utensilios necesarios.

En esta analogía el **programa** es la receta: es un algoritmo que expresado en un lenguaje adecuado indica los pasos a seguir para cocinar el pastel.

El chef es el procesador **CPU**, ya que es el que "procesa" (bate, mezcla, etc.) los ingredientes.

Los ingredientes son los **datos de entrada**.

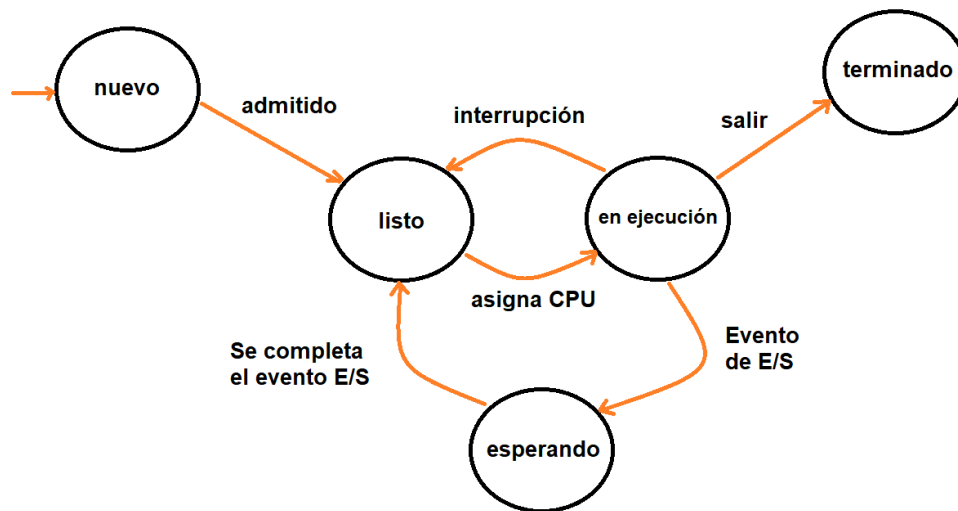
El **proceso** son todas las actividades que realiza el chef: leer e interpretar la receta, mezclar los ingredientes, meter al horno la mezcla, etc.

Durante el tiempo en que el chef está haciendo el pastel puede ocurrir algo que interrumpa su actividad, por ejemplo, que reciba una llamada. En ese momento el chef puede registrar en cuál paso de la receta se quedó: estado del proceso. También en ese momento decide si tomar la llamada o no. Si la toma, al regresar deberá retomar el proceso donde se quedó.

Un procesador real puede ser compartido por muchos procesos, para ellos es necesario contar con un algoritmo planificador (*scheduler*) para determinar cuales procesos ejecutar y cuales detener.

Estados de un proceso

Un proceso es una entidad activa o dinámica, y durante su ciclo de vida puede tomar diferentes estados, como se ve en el siguiente diagrama básico:



Un proceso que está siendo creado está en estado **nuevo**. Una vez que ya se creó entonces pasa a estado **listo**, es decir, solo espera a que se le asigne tiempo de procesador.

Cuando ya se le asignó tiempo de procesador entonces está en **ejecución**. Al estar realizando su trabajo puede ser interrumpido para que el procesador atienda a otros procesos. En este caso el proceso regresa a un estado **listo**. Otro caso sucede cuando al estar en ejecución el proceso pasa a **esperando**. Esto significa que el proceso está a la espera de que ocurra algún evento, p.e. que el usuario teclee enter.

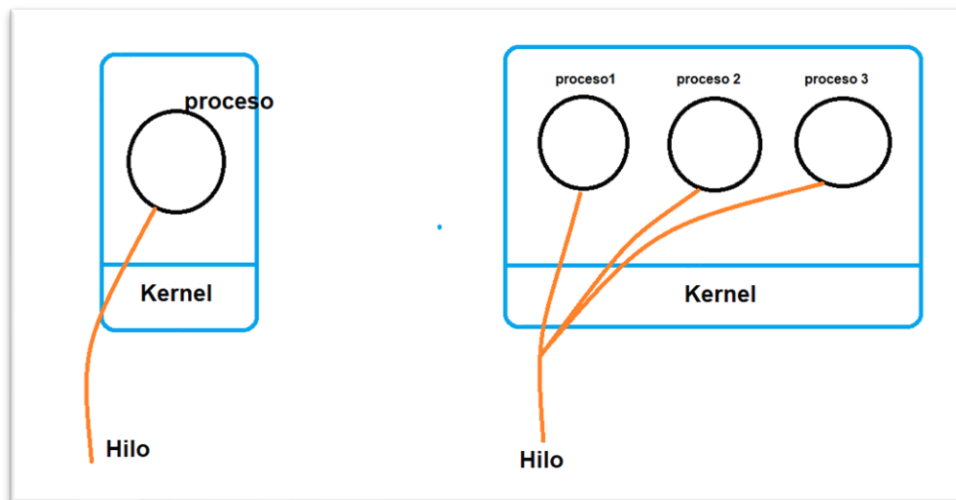
Finalmente, cuando el proceso termina de realizar su tarea entonces pasa a estado **terminado**.

Bloque de control de procesos (process control block – PCB). Es una tabla utilizada por el S.O. para alojar toda la información de cada proceso. Principalmente contiene:

- Estado del proceso.
- Contador de programa. Indica cuál la dirección de la siguiente instrucción a ser ejecutada.
- Registros del CPU. Acumuladores, apuntadores de pila, y otros registros de propósito general.
- Información del CPU-Scheduling. Incluye información sobre la prioridad de un proceso, apuntadores a las colas del scheduling, y cualquier otro parámetro del scheduling.
- Información sobre la administración de la memoria: registros, tablas de segmento, etc.
- Información de cuenta: cantidad de CPU y tiempo real usado, límites tiempo, etc.
- Información sobre estados de E/S: lista de dispositivos de E/S asignados al proceso, lista de archivos abiertos, etc

Hilos

Hasta ahora hemos visto un modelo que implica un proceso para realizar una tarea a la vez: es decir, un solo hilo de ejecución. Este modelo ha sido extendido para poder tener múltiples hilos de ejecución. De esta manera se pueden realizar más de una tarea a la vez. En los sistemas *multicore* se permite que diferentes hilos sean ejecutados en paralelo. Además, el bloque de control de procesos se ha adaptado para que se incluyan todos los hilos.



Un hilo es una unidad básica de utilización del CPU. Esta contiene un solo id para ese hilo, un contador de programa, un conjunto de registros, y su pila (en el TCB). Si más de un hilo parte del mismo proceso entonces compartirán la sección de código, la sección de datos, etc. Veamos una analogía para comprender el modelo de hilos.

Supongamos que un servidor Web acepta solicitudes de los clientes que solicitan páginas Web. Si el servidor tiene muchos clientes y funciona con un solo hilo de ejecución, entonces solo va a poder atender a un solo cliente. Entonces en algún momento los clientes tendrán que esperar mucho tiempo a ser atendidos.

Una solución es que el servidor funcione de tal manera que cuando esté atendiendo una solicitud y reciba otra petición, entonces cree otro proceso para dar servicio a la nueva solicitud. Esto implica que para crear un nuevo cada vez se requerirán muchos recursos. Otra posible solución es el uso de hilos. Esto por lo general es más eficiente que usar un proceso a la vez (por petición). La ventaja de usar varios hilos es que estos se ejecutarán al mismo tiempo en diferentes procesadores o núcleos, mientras que si tenemos varios procesos no se garantiza que se ejecuten al mismo tiempo.